

A Visual and Textual Recurrent Neural Network for Sequential Prediction

Qiang Cui¹ Shu Wu¹ Qiang Liu¹ Liang Wang^{1,2}

¹Center for Research on Intelligent Perception and Computing
National Laboratory of Pattern Recognition

²Center for Excellence in Brain Science and Intelligence Technology
Institute of Automation, Chinese Academy of Sciences

cuiqiang2013@ia.ac.cn, {shu.wu, qiang.liu, wangliang}@nlpr.ia.ac.cn

Abstract

Sequential prediction is a fundamental task for Web applications. Due to the insufficiency of user feedbacks, sequential prediction usually suffers from the cold start problem. There are two kinds of popular approaches based on matrix factorization (MF) and Markov chains (MC) for item prediction. MF methods factorize the user-item matrix to learn general tastes of users. MC methods predict the next behavior based on recent behaviors. However, they have limitations. MF methods can merge additional information to address cold start but could not capture dynamic properties of user's interest, and MC based sequential methods have difficulty in addressing cold start and has a strong Markov assumption that the next state only depends on the last state. In this work, to deal with the cold start problem of sequential prediction, we propose a RNN model adopting visual and textual content of items, which is named as Visual and Textual Recurrent Neural Network (VT-RNN). We can simultaneously learn the sequential latent vectors that dynamically capture the user's interest, as well as content-based representations that contribute to address the cold start. Experiments on two real-world datasets show that our proposed VT-RNN model can effectively generate the personalized ranking list and significantly alleviate the cold start problem.

Introduction

Recently, with the development of Internet, applications with sequential information have become numerous and multilateral, such as web page recommendation and click prediction. Based on sequential prediction methods, these platforms could predict user's next behaviors so as to improve user experience. Taking online shopping as an example, user's purchasing history can be considered as sequences in time order and new items are released continuously. Due to the sparse user feedbacks, sequential prediction often encounters cold start problem. Thus, in this work, our task concentrates on sequential prediction of implicit feedback (online purchase behaviors) and address cold start problem of items with visual and textual information.

As for item prediction, Matrix Factorization (MF) based methods (Koren et al. 2009; Rendle et al. 2009; Salakhutdinov and Mnih 2011) have been widely developed by learning latent representations of users and items. In order to

alleviate the cold start problem, many kinds of additional information have been adopted, such as the attribute information (Lu et al. 2013; Zhao et al. 2013), text (Bao, Fang, and Zhang 2014), images (He and McAuley 2016; Zhao et al. 2016) and so on. However, although these methods can well utilize different types of features to address cold start, they usually capture user's static interest and have difficulty in capturing sequential information. Long-term interest should be weakened while short-term interest would become prominent relatively (Chen, Wang, and Wang 2015).

On the other hand, Markov Chain (MC) based methods (Yang et al. 2010; Rendle, Freudenthaler, and Schmidt-Thieme 2010; Chen, Wang, and Wang 2015) are widely studied for sequential prediction by learning the transition matrix. They predict the next action based on recent actions as the transition matrix gives the probability of different states. However, they have not effectively addressed the cold start problem because the probability is independent of additional information which is crucial to cold start (Koren et al. 2009; Zhao et al. 2016). Further, the Markov assumption prohibits exactly modeling user's long-term interest. MC methods usually consider recent behaviors, which ignores the long-term interest (Shani, Heckerman, and Brafman 2005).

Recently, Recurrent Neural Networks (RNN) has shown great achievements in machine translation (Auli et al. 2013), sequential click prediction (Zhang et al. 2014), location prediction (Liu et al. 2016), next basket recommendation (Yu et al. 2016), multi-behavioral sequential prediction (Liu, Wu, and Wang 2016), and so on. We can see that RNN methods are more promising than Factorizing Personalized Markov Chains (FPMC) (Rendle, Freudenthaler, and Schmidt-Thieme 2010) and other traditional MC methods. However, these existing work have difficulty in incorporating feature information to alleviate the cold start problem.

In view of the above analyses, we propose a model called Visual and Textual Recurrent Neural Network (VT-RNN) for sequential prediction and addressing cold start of items. Intuitively, a user u would look into images and corresponding textual descriptions before he or she buys an item (e.g., sweater), because such kind of knowledge is really beneficial to understand the item and satisfy the user's preference. These additional information should not be ignored for prediction. First we gain the powerful visual and textual fea-

tures from item images and their descriptions respectively. Then we obtain a joint representation of latent, visual and textual features for each item, which is used as the input layers in VT-RNN. Next VT-RNN adopts recurrent structure to capture the dynamic changes of user’s interest represented by hidden layers. Compared to MC in theory, this structure can hold the long-term interest as well as strengthen the recent item’s effect on short-term interest. Then, a Bayesian Personalized Ranking (BPR) framework (Rendle et al. 2009) and the Back Propagation Through Time (BPTT) algorithm (Werbos 1990) are used to learn parameters, because BPR is state-of-the-art on implicit feedback. The main contributions are listed as follows:

- We propose a recurrent model that can incorporate visual and textual information to make sequential prediction, which can well reveal item’s characteristics and user’s dynamic preference.
- We design a novel experiment to indicate that visual and textual information could significantly alleviate the cold start of items in sequential prediction.
- Experiments on two large real-world datasets reveal that VT-RNN is very effective and outperforms the state-of-the-art methods.

Related Work

In this section, we review several related works including matrix factorization methods (MF), bayesian personalized ranking methods (BPR), Markov Chain methods (MC), and recurrent neural networks (RNN).

Matrix factorization based methods are one of the most fundamental and popular approaches for recommendation. The basic idea of MF is to use two low-rank matrices to approximate the original user-item matrix. Recently, MF absorbs rich additional information to alleviate the cold start problem, like item’s categories or user’s demographics (Lu et al. 2013; Zhao et al. 2013). Textual information such as reviews is used (Levi et al. 2012) along with the development of online shopping. Zhao et al. (2016) extend the MF by combining visual data like posters and still frames of a movie to understand movie and user’s interest. However, none of these methods could reflect the changes of user’s preference over time.

In recent years, pairwise methods are eye-catching because it is powerful on implicit feedback. These methods can directly optimize the ranking of feedbacks because positive items are preferable to negative items. Rendle et al. (2009) propose a bayesian personalized ranking (BPR) framework and gain the best performance by maximizing the user’s preference for positive and negative items. Recently, BPR is extended to combine more information like users’ social relations (Zhao, McAuley, and King 2014). Other information like visual signals is accommodated by VBPR (He and McAuley 2016), which applies visual features of product images to uncover additional users’ visual preference and better understand items. Similar to MF methods, they learn general tastes of users.

In addition to MF based methods above, sequential methods are very popular, like traditional sequential recommen-

dations mostly rely on Markov chains. Rendle, Freudenthaler, and Schmidt-Thieme (2010) make a combination of MF and MC to learn both general taste and current effect for next-basket recommendation. Yang et al. (2010) propose a topic-aware Markov model to predict the next web page when a user surfs on the internet. Chen, Wang, and Wang (2015) build a Markov model integrated with the forgetting mechanism to weaken long-term interest and highlight short-term interest for item recommendation. However, they have difficulty in resolving the cold start.

Recently, sequential recurrent neural network is getting more and more powerful. RNN has input layers, hidden layers and output layers. Owing to its recurrent structure, RNN can better extract the dependencies among different sequences. RNN based sequential click prediction (Zhang et al. 2014) gains state-of-the-art performance. Yu et al. (2016) takes the representation of a basket acquired by pooling operation as the input layer of RNN, which is most effective for next basket recommendation. Liu et al. (2016) incorporates time-specific and distance-specific transition matrices in RNN to predict next location. Liu, Wu, and Wang (2016) combines RNN and Log-BiLinear model (Mnih and Hinton 2007) to make multi-behavioral prediction. Compared with traditional sequential methods, RNN is more promising.

Proposed VT-RNN Model

In this section, we construct a Visual and Textual Recurrent Neural Network model (VT-RNN). We first formulate the problem and build the item representation with visual and textual features, then introduce the VT-RNN model in detail. Finally, we show the training procedure with the Bayesian Personalized Ranking (BPR) framework and the Back Propagation Through Time (BPTT) algorithm.

Problem Formulation

We focus on sequential prediction of purchase histories from online shopping. Let $U = \{u_1, \dots, u_{|U|}\}$ and $I = \{i_1, \dots, i_{|I|}\}$ represent the sets of users and items respectively. Use $I^u = (i_1^u, \dots, i_m^u)$ to denote the positive items that user u has purchased in chronological order with t -th item $i_t^u \in I$. Additionally, a single image and a textual description are available for each item $i \in I$ in our datasets. Given each user’s history I^u , our goal is to predict a ranking list of new items ($I \setminus I^u$) that the user would purchase in the future, because it is not necessary to predict items that user has already bought. The notation of this work is listed in Table 1 for clarity.

New Representation of Item

Sequential prediction often encounters cold start problem as the feedbacks are too sparse to learn fine representations of users and items. One effective way to alleviate this situation is to merge additional information like user attributes (Koren et al. 2009). In this work, we construct a recurrent model to make sequential prediction with visual and textual features.

The directly observable visual and textual features (f and g) are obtained from CNN (Szegedy et al. 2015) and GloVe (Pennington, Socher, and Manning 2014) weighted by TF-IDF respectively. The two kinds of features have 1024 and

Table 1: Notation

Notation	Explanation
U, I	set of users/items
I^u, m	sequence and number of training items of a user u
T^u	sequence of test items of a user u
p, q	positive item, negative item
\hat{x}_{upq}^t	preference of user u towards item p and q at t -th input
d	dimension of latent features
$\mathbf{x}, \mathbf{f}, \mathbf{g}$	latent, visual and textual features of an item
\mathbf{E}, \mathbf{V}	embedding matrices for visual and textual features
\mathbf{U}, \mathbf{W}	transition matrices for recurrent neural network
\mathbf{i}, \mathbf{h}	input and hidden layers of recurrent neural network

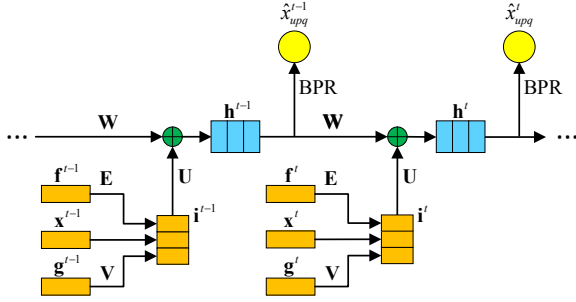


Figure 1: Diagram of VT-RNN model. The input is composed of latent, visual and textual features of item and we concatenate them as the input to represent the item. The hidden layer captures dynamic information to represent the user’s interest. We use the BPR framework and the BPTT algorithm to train the proposed model.

100 dimensions respectively. The natural solution to build our model is to directly use these features. However, if we directly use the original representation, it will lead to a high computation complexity. Therefore, we learn two embedding matrices \mathbf{E} and \mathbf{V} to transform the original high-dimensional features to embedded low-dimensional features denoted by $\mathbf{i}_f = \mathbf{E}\mathbf{f}$ and $\mathbf{i}_g = \mathbf{V}\mathbf{g}$.

As a complement, we introduce indirectly observable latent feature $\mathbf{i}_x = \mathbf{x}$ to indicate the information that could not be represented by visual and textual features. Three kind of features have the same d -dimensional length. Then, we concatenate them together as the new representation of item:

$$\mathbf{i} = [\mathbf{i}_x; \mathbf{i}_f; \mathbf{i}_g] = [\mathbf{x}; \mathbf{E}\mathbf{f}; \mathbf{V}\mathbf{g}] \quad (1)$$

Please note that subscript x, f and g represent three equal length parts of a vector which denote the latent, visual and textual features respectively in our work.

Visual and Textual Recurrent Neural Network

In this section, we introduce our VT-RNN model in detail as shown in Figure 1. VT-RNN has input layers, hidden layers, output layers and inner matrices, similar to general RNN.

As we have obtained the representation of item used as input layers, then we obtain the full user representation which also has latent, visual and textual features by computing hid-

den layer vector.

$$\begin{aligned} \mathbf{h}^t &= [\mathbf{h}_x^t; \mathbf{h}_f^t; \mathbf{h}_g^t] \\ &= \sigma(\mathbf{U}\mathbf{i}_p^t + \mathbf{W}\mathbf{h}^{t-1}) \\ &= \sigma(\mathbf{U}[\mathbf{x}_p^t; \mathbf{E}\mathbf{f}_p^t; \mathbf{V}\mathbf{g}_p^t] + \mathbf{W}\mathbf{h}^{t-1}) \end{aligned} \quad (2)$$

where \mathbf{h}^t is used as the user representation at t -th input activated by *sigmoid* function $\sigma(x) = \frac{1}{1+e^{-x}}$. \mathbf{i}_p^t denotes the positive input and we omit the subscript p in Figure 1. Transition matrix \mathbf{U} transfers the input to modify the user’s current preference. The recurrent connection \mathbf{W} delivers the previous sequential information (e.g., user’s purchases).

After discussing the input and hidden layers of the VT-RNN model, we introduce BPR (Rendle et al. 2009) framework to compute user’s preference as the output layers. BPR is a powerful pairwise method for implicit feedback, and it has been successfully used to learn RNN based models (Yu et al. 2016; Liu et al. 2016; Liu, Wu, and Wang 2016).

The training set S is a triple of (u, p, q) , where u represents the user, p and q denote the positive and negative items respectively. Item p is selected one by one from user’s purchase history I^u , while item q is randomly chosen from the items $(I \setminus I^u)$ that user u have not purchased.

$$S = \{(u, p, q) | u \in U \wedge p \in I^u \wedge q \in I \setminus I^u\} \quad (3)$$

Given the training set, we need a predictor to compute the user’s preference for positive and negative items. The preference at the t -th input can be computed easily using dot product:

$$\begin{aligned} \hat{x}_{upq}^t &= \hat{x}_{up}^t - \hat{x}_{uq}^t \\ &= (\mathbf{h}^{t-1})^T (\mathbf{i}_p^t - \mathbf{i}_q^t) \end{aligned} \quad (4)$$

where \mathbf{h}^{t-1} is the previous hidden layer. \mathbf{i}_p^t and \mathbf{i}_q^t represents the current positive and randomly selected negative input at t -th input respectively.

The objective function is a maximum posterior estimator

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}} \sum_{(u,p,q) \in S} \ln \sigma(\hat{x}_{upq}) - \frac{\lambda_\Theta}{2} \|\Theta\|^2 \quad (5)$$

and $\lambda_\Theta \geq 0$ are regulation parameters, Θ denotes the vector parameters $\Theta = \{\mathbf{U}, \mathbf{W}, \mathbf{X}, \mathbf{E}, \mathbf{V}\}$ and \mathbf{X} is the set of latent feature \mathbf{x} . Then, the model can be learned by the stochastic gradient ascent approach and the parameters are updated in the following way:

$$\Theta = \Theta + \alpha \left(\sigma(-\hat{x}_{upq}) \frac{\partial \hat{x}_{upq}}{\partial \Theta} - \lambda_\Theta \Theta \right) \quad (6)$$

where α is the learning rate.

Parameter Inference

In this section, we employ the Back Propagation Through Time (BPTT) (Werbos 1990) to learn the parameters. As RNN has feedforward and feedback path, the parameters would be updated in two directions.

In the forward direction, we update the positive and negative items at t -th input in Eq. 4 by $\partial \hat{x}_{upq} / \partial \Theta = (\partial \hat{x}_{upq} / \partial (\mathbf{i}_p^t - \mathbf{i}_q^t)) (\partial (\mathbf{i}_p^t - \mathbf{i}_q^t) / \partial \Theta)$. Their derivatives are

$$\frac{\partial \hat{x}_{upq}}{\partial \mathbf{x}_p^t} = \mathbf{h}_x^{t-1} \quad (7)$$

$$\frac{\partial \hat{x}_{upq}}{\partial \mathbf{x}_q^t} = -\mathbf{h}_x^{t-1} \quad (8)$$

$$\frac{\partial \hat{x}_{upq}}{\partial \mathbf{E}} = \mathbf{h}_f^{t-1} (\mathbf{f}_p^t - \mathbf{f}_q^t)^T \quad (9)$$

$$\frac{\partial \hat{x}_{upq}}{\partial \mathbf{V}} = \mathbf{h}_g^{t-1} (\mathbf{g}_p^t - \mathbf{g}_q^t)^T \quad (10)$$

Then we can easily update these parameters according to Eq. 6. Note that superscript t ranges from 2 to m .

In the backward direction, we update the user representation (hidden layer) at $(t-1)$ -th input in Eq. 4 by $\partial \hat{x}_{upq} / \partial \Theta = (\partial \hat{x}_{upq} / \partial (\mathbf{h}^{t-1})) (\partial (\mathbf{h}^{t-1}) / \partial \Theta)$. Their derivatives are

$$\frac{\partial \hat{x}_{upq}}{\partial \mathbf{U}} = \mathbf{m} (\mathbf{i}_p^{t-1})^T \quad (11)$$

$$\frac{\partial \hat{x}_{upq}}{\partial \mathbf{W}} = \mathbf{m} (\mathbf{h}^{t-2})^T \quad (12)$$

$$\frac{\partial \hat{x}_{upq}}{\partial \mathbf{x}_p^{t-1}} = (\mathbf{U}^T \mathbf{m})_x \quad (13)$$

$$\frac{\partial \hat{x}_{upq}}{\partial \mathbf{E}} = (\mathbf{U}^T \mathbf{m})_f (\mathbf{f}_p^{t-1})^T \quad (14)$$

$$\frac{\partial \hat{x}_{upq}}{\partial \mathbf{V}} = (\mathbf{U}^T \mathbf{m})_g (\mathbf{g}_p^{t-1})^T \quad (15)$$

where $\mathbf{m} = (\mathbf{i}_p^t - \mathbf{i}_q^t) \odot ((\vec{1} - \mathbf{h}^{t-1}) \odot \mathbf{h}^{t-1})$ and \odot means the element-wise product between two vectors. This process would be repeated iteratively at each hidden layer and these gradients come from the BPR process. Due to the recurrent structure, other gradients iteratively propagate from the current hidden layer to previous layer. Two parts of gradients combine together at hidden layer and we regard gradients at input terminal of \mathbf{h}^t as \mathbf{e}^t . Then we can use BPTT to update all parameters in the backward direction.

$$\mathbf{U} = \mathbf{U} + \alpha \left(\sum_{z=1}^{m-1} \mathbf{e}^{m-z} (\mathbf{i}_p^{m-z})^T - \lambda \mathbf{U} \right) \quad (16)$$

$$\mathbf{W} = \mathbf{W} + \alpha \left(\sum_{z=1}^{m-1} \mathbf{e}^{m-z} (\mathbf{h}_p^{m-z-1})^T - \lambda \mathbf{W} \right) \quad (17)$$

$$\mathbf{x}_p^{t-1} = \mathbf{x}_p^{t-1} + \alpha ((\mathbf{U}^T \mathbf{e}^{t-1})_x - \lambda \mathbf{x}_p^{t-1}) \quad (18)$$

$$\mathbf{E} = \mathbf{E} + \alpha \left(\sum_{z=1}^{m-1} (\mathbf{U}^T \mathbf{e}^{m-z})_f (\mathbf{f}_p^{m-z})^T - \lambda_e \mathbf{E} \right) \quad (19)$$

$$\mathbf{V} = \mathbf{V} + \alpha \left(\sum_{z=1}^{m-1} (\mathbf{U}^T \mathbf{e}^{m-z})_g (\mathbf{g}_p^{m-z})^T - \lambda_v \mathbf{V} \right) \quad (20)$$

Instead of updating the matrices \mathbf{U} , \mathbf{W} , \mathbf{E} and \mathbf{V} from one hidden layer to its former hidden layer step by step, the BPTT algorithm adds all the loss throughout the sequence and updates the parameters in one step. Note that the superscript t also ranges from 2 to m . And we introduce λ_e and λ_v to regularize embedding matrices \mathbf{E} and \mathbf{V} respectively.

Table 2: Preprocessed Datasets

Dataset	#users $ U $	#items $ I $	feedback	sparsity (%)
TaoBao	37,690	282,759	1,873,429	99.982
Amazon	41,878	355,571	678,329	99.995

Experimental Results and Analysis

In this section, we conduct experiments on two real-world datasets. Both datasets have massive sequential implicit feedback, as well as provide one image and a textual description for each item.

Experimental Settings

Datasets Experiments are conducted on two large datasets namely Taobao¹ and Amazon². The basic statistics of them are listed in Table 2. We select the first 90% sequential histories as training set and the rest 10% as test set.

- **TaoBao** is the dataset for clothing matching competition on *TianChi*³ platform. We use user historical behavior data and item features (image, text) to make sequential prediction. We hold users who purchase at least 30 times ($|I^u| \geq 30$).
- **Amazon** contains many reviews and product metadata (McAuley et al. 2015; McAuley, Pandey, and Leskovec 2015). We use one large category *Clothing*, *Shoes* and *Jewelry* for experiments. We acquire the sequential implicit feedback from review histories, obtain the images and text data from product metadata. We choose users with more than 10 items ($|I^u| \geq 10$).

Visual Features We obtain the visual features by using the GoogLeNet model (Szegedy et al. 2015) implemented by BVLC Caffe deep learning framework (Jia et al. 2014). The network has 22 layers and has been pre-trained on 1.2M ImageNet ILSVRC2012 images (Russakovsky et al. 2015) in Caffe. In our experiments, we use the output of layer 'pool5/7x7_s1' to obtain 1024-dimensional visual features normalized to range [0, 0.5] by min-max normalization.

Textual Features On TaoBao, we directly use the item title which has already been segmented and disordered. On Amazon, we combine each item's category and title as the text data. Thus we adopt the GloVe model (Pennington, Socher, and Manning 2014) weighted by TF-IDF (Salton and Buckley 1988) to obtain each word's feature and weight. Finally, we compute the weighted features for all items to obtain 100-dimensional textual features normalized to range [-0.5, 0.5] by min-max normalization.

Evaluation Metrics We evaluate the performance in the test set using metrics including Recall, Precision, Mean Average Precision (MAP) (Manning, Raghavan, and Schütze 2008) and Normalized Discounted Cumulative Gain (NDCG) (Wang et al. 2015). The former two are evaluations of un-ranked retrieval sets while the latter two are evaluations of

¹<https://tianchi.shuju.aliyun.com/datalab/dataSet.htm?id=13>

²<http://jmcauley.ucsd.edu/data/amazon/>

³<https://tianchi.shuju.aliyun.com/>

Table 3: Evaluation of different methods on two datasets with dimension $d = 20$. We recommend top-10, 30 and 50 items for each user. AUC is listed in the last column.

dataset	method	@10				@30				@50				AUC
		Recall	Precision	MAP	NDCG	Recall	Precision	MAP	NDCG	Recall	Precision	MAP	NDCG	
TaoBao	Random	0.0032	0.0013	0.0013	0.0027	0.0120	0.0017	0.0011	0.0048	0.0152	0.0013	0.0028	0.0077	0.5000
	POP	0.0310	0.0130	0.0132	0.0273	0.1818	0.0254	0.0392	0.1057	0.4940	0.0414	0.0387	0.1951	0.6147
	MF	0.0760	0.0318	0.0226	0.0535	0.3471	0.0485	0.0639	0.1877	0.6296	0.0527	0.0668	0.2663	0.6633
	BPR	0.3281	0.1374	0.0895	0.2232	0.8342	0.1165	0.1061	0.3779	1.3143	0.1101	0.1154	0.4945	0.6901
	VBPR	0.2952	0.1236	0.0843	0.2056	0.7740	0.1081	0.1017	0.3503	1.1933	0.1000	0.1009	0.4507	0.7310
	T-BPR	0.3021	0.1266	0.0851	0.2046	0.7955	0.1111	0.1043	0.3556	1.2383	0.1037	0.1083	0.4706	0.7477
	VT-BPR	0.3211	0.1345	0.0950	0.2257	0.8291	0.1158	0.1070	0.3775	1.2972	0.1087	0.1209	0.5042	0.7659
	RNN	0.3705	0.1552	0.1027	0.2532	0.9070	0.1266	0.1071	0.4030	1.3750	0.1152	0.1069	0.5119	0.6574
	V-RNN	0.4706	0.1971	0.1336	0.3312	1.1939	0.1667	0.1548	0.5502	1.8400	0.1542	0.1565	0.6986	0.7778
	T-RNN	0.3794	0.1589	0.1246	0.2875	1.0134	0.1415	0.1308	0.4657	1.5607	0.1308	0.1315	0.6022	0.8090
	VT-RNN	0.4795	0.2008	0.1441	0.3451	1.2535	0.1750	0.1613	0.5877	1.8710	0.1568	0.1554	0.7181	0.8262
Amazon	Random	0.0034	0.0007	0.0019	0.0029	0.0045	0.0003	0.0006	0.0018	0.0102	0.0004	0.0006	0.0030	0.5000
	POP	0.2394	0.0504	0.0660	0.1290	0.5048	0.0354	0.0612	0.1820	0.7328	0.0309	0.0554	0.2189	0.5490
	MF	0.1055	0.0222	0.0323	0.0593	0.2609	0.0183	0.0298	0.0928	0.3823	0.0161	0.0322	0.1172	0.5092
	BPR	0.0998	0.0210	0.0307	0.0564	0.2394	0.0168	0.0312	0.0870	0.3335	0.0140	0.0276	0.0998	0.5092
	VBPR	0.1475	0.0310	0.0438	0.0833	0.3222	0.0226	0.0374	0.1152	0.4458	0.0188	0.0358	0.1359	0.7143
	T-BPR	0.0760	0.0160	0.0236	0.0436	0.2609	0.0183	0.0459	0.1079	0.4220	0.0178	0.0415	0.1335	0.7607
	VT-BPR	0.1202	0.0253	0.0340	0.0647	0.3573	0.0251	0.0482	0.1335	0.5524	0.0233	0.0464	0.1653	0.7680
	RNN	0.2496	0.0525	0.0672	0.1334	0.5116	0.0359	0.0598	0.1818	0.7453	0.0314	0.0565	0.2227	0.5671
	V-RNN	0.2802	0.0590	0.0733	0.1472	0.6852	0.0481	0.0829	0.2437	0.9960	0.0419	0.0729	0.2970	0.7310
	T-RNN	0.2484	0.0523	0.0749	0.1418	0.6001	0.0421	0.0811	0.2228	0.9064	0.0382	0.0820	0.2870	0.7762
	VT-RNN	0.3120	0.0657	0.0860	0.1667	0.7101	0.0498	0.0955	0.2642	1.0482	0.0441	0.0864	0.3168	0.7896

ranked retrieval results. Here we consider top- k (e.g., 10, 30 and 50) recommendation. Besides, we introduce the Area Under the ROC Curve (AUC) (Rendle et al. 2009) to evaluate the overall performance.

Analogous to settings in Rendle, Freudenthaler, and Schmidt-Thieme (2010), only new items for user u are considered. We delete the items from each user’s test sequence T^u that also appear in training sequence I^u . Besides, we remove the duplicate items in T^u so that items in T^u are all uniquely new items for user u . Note that this setting leads to a low top- k performance because it is more difficult to recommend new items than already bought ones.

We compare VT-RNN with several comparative methods:

- Random: Items are randomly ranked for all users.
- POP: This baseline recommends the most popular items in the training set for each user u .
- MF: This method maps the interaction between users and items to a latent space of lower dimension.
- BPR: This pairwise method refers to the BPR-MF proposed by Rendle et al. (2009) for implicit feedback.
- VBPR: Introduced by He and McAuley (2016), this is an extended method with visual features based on BPR-MF.
- RNN: This sequential baseline is presented in Liu et al. (2016) and outperforms the MC based methods.

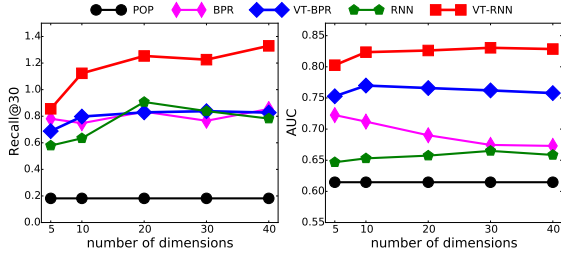
Models we propose consist of two parts. First part is based on BPR. We use textual features to replace visual features in VBPR to obtain the BPR+Text, abbreviated as T-BPR. We add textual features to VBPR to obtain the VBPR+Text,

abbreviated as VT-BPR. Second, we add visual and textual features to RNN step by step. RNN+Image, RNN+Text and RNN+Image+Text are abbreviated as V-RNN, T-RNN and VT-RNN, respectively. Additionally, we need to initialize the parameters to the same range, e.g., uniform distribution $[-0.5, 0.5]$. Besides, the learning rate is set as $\alpha = 0.1$ for BPR based methods and RNN based methods while $\alpha = 0.01$ for MF, regularizers are set as $\lambda_\Theta = 0.001$.

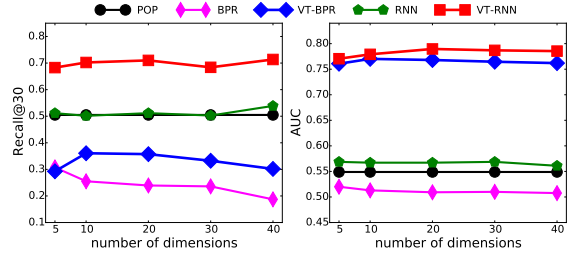
Analysis of Experimental Results

Table 3 illustrates all the performance comparison on two datasets with five evaluation metrics. All values of evaluation metrics should multiply 0.01 to obtain the real ones in Table 3, Figures 2 and 3 except AUC.

First, from a global perspective, additional visual and textual information of items is indeed helpful. VT-BPR beats other BPR models. VT-BPR beats other RNN models and almost gains the best performance. These information can be effectively modeled by our VT-RNN. *Second*, with the help of more content information (visual and textual), models can obtain the further enhancement. V-RNN and T-RNN outperform RNN, and VT-RNN performs the best. For example, on two datasets, VT-RNN beats RNN by 29.42% and 18.15% on Recall@10, 40.31% and 27.83% on MAP@10, 25.68% and 38.62% on AUC respectively. These highlight the effectiveness of multiple information. *Third*, recurrent structure can capture sequential information which is help for prediction. Both trained by BPR framework, basic RNN and VT-RNN beats corresponding basic BPR and VT-BPR respectively. All these indicate that VT-RNN has the higher applicability because it has the best and stable performance

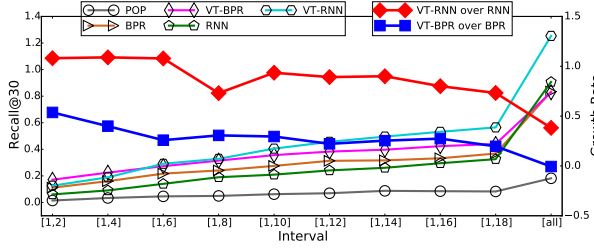


(a) TaoBao

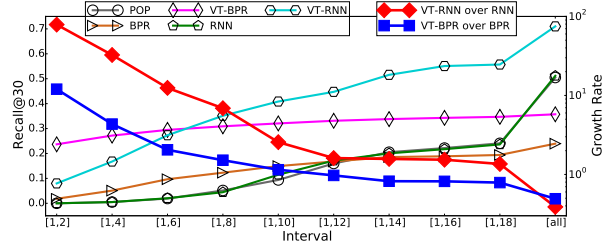


(b) Amazon

Figure 2: Recall@30 and AUC performance of VT-RNN on two datasets with varied dimensions.



(a) TaoBao.Cold.Start



(b) Amazon.Cold.Start

Figure 3: Cold start analysis on two datasets with dimension $d = 20$. Five curves of Recall@30 and two curves of growth rate (degree of improvement) of VT-BPR over BPR and VT-RNN over RNN are shown on left and right vertical axes respectively.

on different datasets and can finely model the visual and textual information for the personalized ranking.

Dimensionality Analysis

The dimensionality analysis is investigated in Figure 2. We illustrate the performance with Recall@30 and AUC on two datasets. We leave out the Precision, MAP and NDCG as they have the similar trend with Recall. The best dimension is set up to $d = 20$. Generally, it is obvious that VT-RNN outperforms all the other methods on all dimensionalities with a large improvement. On the other hand, with the increasing of dimensionality, VT-RNN still has the much stable performance which demonstrates the ability to prevent over-fitting. We conclude that VT-RNN has a higher stability because it is not sensitive to dimensionality.

Analysis of Cold Start

In this section, we analyse the cold start in detail. Items in test set are divided into 10 bins (e.g., $[1, 2]$, $[1, 4]$) according to their frequency. For example, bin $[1, 2]$ has the items that appear in the test set for 1 and 2 times. Note that the smaller the bin is, the higher proportion of cold start items it has. Last bin $[all]$ represents the whole test set. Then we calculate the Recall@30 on each bin. Finally, we compute the growth rate (degree of improvement) of VT-BPR and VT-RNN over corresponding BPR and RNN on each bin respectively.

Figure 3 illustrates five Recall@30 performance curves and two growth rate curves on left and right vertical axes respectively. For Recall@30 curves, VT-RNN has the best performance on most bins. Most models have a large improvement from bin $[1, 18]$ to bin $[all]$, because it is easier to predict much frequent items, especially for VT-RNN.

Specially, we focus on the two growth rate curves. For a global perspective, as the bin has larger proportion of cold start items (seeing from right side of figure to left side of figure), growth rate becomes larger and larger. This indicates that visual and textual information are indeed beneficial to relieve the cold start. Further, when cold start problem gets worse on bins with bigger proportion of cold start items, visual and textual information can much significantly relieve the problem. Because cold start items have few interactions with users and other items, directly related visual and textual information would effectively represent the item’s characteristics and satisfy user’s preference. On the other hand, growth rate is much larger on Amazon than Taobao. On Taobao, VT-RNN has about 100% growth rate over RNN on small bins (e.g., $[1, 6]$, $[1, 4]$ and $[1, 2]$). On Amazon, VT-RNN has about 1.25×10^3 %, 3.25×10^3 % and 7.88×10^3 % growth rates over RNN on bins $[1, 6]$, $[1, 4]$ and $[1, 2]$ respectively. Because Amazon is sparser than Taobao, which is shown in Table 2, so that the cold start problem is worse. Then we can see that visual and textual information are more powerful on dataset with worse cold start problem.

Visual and textual information are much helpful to alleviate the cold start and VT-RNN is very effective. We believe VT-RNN can better capture the sequential information as well as content information (visual and textual).

Conclusion

In this work, we have proposed a novel visual and textual model based on recurrent neural networks (VT-RNN). We construct comprehensive item and user representations with latent, visual and textual features. The transformation of user’s preference is leveraged in a recurrent structure. BPR

and BPTT are used to train our model. Experiments validate the state-of-the-art performance of our VT-RNN. Visual and textual information could indeed significantly alleviate the cold start problem.

References

- [2013] Auli, M.; Galley, M.; Quirk, C.; and Zweig, G. 2013. Joint language and translation modeling with recurrent neural networks. In *EMNLP*, volume 3, 0.
- [2014] Bao, Y.; Fang, H.; and Zhang, J. 2014. Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In *AAAI*, 2–8.
- [2015] Chen, J.; Wang, C.; and Wang, J. 2015. A personalized interest-forgetting Markov model for recommendations. In *AAAI*, 16–22.
- [2016] He, R., and McAuley, J. 2016. Vbpr: visual bayesian personalized ranking from implicit feedback. In *AAAI*.
- [2014] Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; and Darrell, T. 2014. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, 675–678. ACM.
- [2009] Koren, Y.; Bell, R.; Volinsky, C.; et al. 2009. Matrix factorization techniques for recommender systems. *Computer* 42(8):30–37.
- [2012] Levi, A.; Mokryn, O.; Diot, C.; and Taft, N. 2012. Finding a needle in a haystack of reviews: cold start context-based hotel recommender system. In *RecSys*, 115–122. ACM.
- [2016] Liu, Q.; Wu, S.; Wang, L.; and Tan, T. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *AAAI*.
- [2016] Liu, Q.; Wu, S.; and Wang, L. 2016. Multi-behavioral sequential prediction for collaborative filtering. *arXiv preprint arXiv:1608.07102*.
- [2013] Lu, Z.; Pan, W.; Xiang, E. W.; Yang, Q.; Zhao, L.; and Zhong, E. 2013. Selective transfer learning for cross domain recommendation. In *SDM*, 641–649. SIAM.
- [2008] Manning, C. D.; Raghavan, P.; and Schütze, H. 2008. *Introduction to information retrieval*. Cambridge University Press.
- [2015] McAuley, J.; Targett, C.; Shi, Q.; and van den Hengel, A. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*, 43–52. ACM.
- [2015] McAuley, J.; Pandey, R.; and Leskovec, J. 2015. Inferring networks of substitutable and complementary products. In *ACM SIGKDD*, 785–794. ACM.
- [2007] Mnih, A., and Hinton, G. 2007. Three new graphical models for statistical language modelling. In *ICML*, 641–648. ACM.
- [2014] Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, 1532–43.
- [2009] Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, 452–461.
- [2010] Rendle, S.; Freudenthaler, C.; and Schmidt-Thieme, L. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *WWW*, 811–820. ACM.
- [2015] Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *IJCV* 115(3):211–252.
- [2011] Salakhutdinov, R., and Mnih, A. 2011. Probabilistic matrix factorization. In *NIPS*, volume 20, 1–8.
- [1988] Salton, G., and Buckley, C. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management* 24(5):513–523.
- [2005] Shani, G.; Heckerman, D.; and Brafman, R. I. 2005. An mdp-based recommender system. *Journal of Machine Learning Research* 6(Sep):1265–1295.
- [2015] Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *CVPR*, 1–9.
- [2015] Wang, P.; Guo, J.; Lan, Y.; Xu, J.; Wan, S.; and Cheng, X. 2015. Learning hierarchical representation model for nextbasket recommendation. In *SIGIR*, 403–412. ACM.
- [1990] Werbos, P. J. 1990. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE* 78(10):1550–1560.
- [2010] Yang, Q.; Fan, J.; Wang, J.; and Zhou, L. 2010. Personalizing web page recommendation via collaborative filtering and topic-aware Markov model. In *ICDM*, 1145–1150. IEEE.
- [2016] Yu, F.; Liu, Q.; Wu, S.; Wang, L.; and Tan, T. 2016. A dynamic recurrent model for next basket recommendation. In *SIGIR*, 729–732. ACM.
- [2014] Zhang, Y.; Dai, H.; Xu, C.; Feng, J.; Wang, T.; Bian, J.; Wang, B.; and Liu, T.-Y. 2014. Sequential click prediction for sponsored search with recurrent neural networks. In *AAAI*, 1369–1376.
- [2013] Zhao, L.; Pan, S. J.; Xiang, E. W.; Zhong, E.; Lu, Z.; and Yang, Q. 2013. Active transfer learning for cross-system recommendation. In *AAAI*. Citeseer.
- [2016] Zhao, L.; Lu, Z.; Pan, J.; Yang, Q.; Xu, W.; Lee, Y.-K.; Gao, B.; Ma, S.; Zhang, A.; Mondal, S. K.; et al. 2016. Matrix factorization+ for movie recommendation. *IJCAI 2016, New York City, USA* 1.
- [2014] Zhao, T.; McAuley, J.; and King, I. 2014. Leveraging social connections to improve personalized ranking for collaborative filtering. In *CIKM*, 261–270. ACM.